

Writing Good Bug Reports

By Robert Spotswood

It's not just for software!

In the software development world, developer time is limited. Having to slog through a badly written bug report takes time and energy many aren't willing to spend. A good bug report, on the other hand, makes seeing what's wrong easy for the developer. This greatly increases the odds of getting the bug fixed.

Unfortunately, even the best written bug reports may not result in a fix. The developers, or their managers, may feel their time is better spent elsewhere. The bug might also be a design trade-off. Still, it's better to have the odds in your favor than not.

When many computer users think of bug reports, they think of reporting software flaws. While this one use for bug reports, it is not the only one. Even if you are never involved at all in software development, knowing (and using!) the elements of good bug reporting can help you throughout life, not just with a computer. A good bug report can mean the difference between the problem being solved quickly and a time consuming, expensive, frustrating experience.

What are some other situations that a good bug report is needed, you ask? Tech support is one: Phone support is one of the hardest types of tech support to do. As many phone techs have quotas to meet, being able to make a good bug report can mean the difference between getting your problem fixed and a "We don't support that." because the phone tech has had to waste so much time trying to figuring out what the problem actually is.

But the usefulness of a good bug report doesn't end with computers. Your author has found the principles involved in a good bug report have also helped in dealing with plumbers, auto mechanics, even doctors. In the case of doctors, a quick and accurate diagnosis thanks to a good bug report could very well mean the difference between life and death, or worse. Anytime you need someone else to help solve a problem for you, they need to understand what the problem is before a good solution is possible, unless they get really lucky. Don't count on luck.

Before You Write a Bug Report

Getting back to the case of computer problems, there are several things you should try before you begin writing a bug report. First, try a reboot. Pulling the plug or removing the battery can also effectively reboot some devices, but only do that if there is no other way to reboot. Sometimes the systems get into an unstable state and reboot will fix the problem. This is such a quick and easy test, it should be one of the first things you try before you start writing a bug report. In fact, HAL-PC & HALNet Tech Support requires that you reboot *before* you call them for help. It also works for non-computers sometimes. This has worked with projectors, external modems (DSL and Cable), copiers, cell phones, and network switches as well, just to name a few.

In the software world, at least, be sure you are up to date. Do you have the latest version of the software, if feasible? -- This may not be realistic in the commercial software arena. Are there any updates/patches to the version you have that you haven't applied? How about any updates/patches to your operating system? As bugs are discovered, patches (aka updates aka service packs) are released with fixes. Someone else could have already discovered the bug, made a good bug report, and the fix is out there waiting for you to use it. As an added bonus, it could fix other bugs you haven't run into yet.

Is there an FAQ ("Frequently Asked Questions") for the product? FAQ's are a good place to look for problems and their solutions. Developers and support people put out FAQ's to answer some common questions. It may be that a fix or workaround for your problem is included. You may also discover new features that you didn't know the software had in the process.

Do a basic system check. Sometimes problems in a piece of software may not be a problem with the software itself, but with the system it runs on. As an example, I have had occasion reports of various problems with Thunderbird from people who still use Windows 98. In about 90-95% of the cases, a scandisk on the hard drive found and fixed errors on the hard drive. After that, the problems disappeared. Knoppix (http://www.hal-pc.org/journal/2004/04_aug/linux.html) is great for testing if it is a hardware issue and not a software issue, and even includes a memory tester, although the memory tester really tests memory, motherboard, CPU and power supply.

Try uninstalling and reinstalling the application. Sometimes things get messed up in the day to day use. Library files accidentally get overwritten, or perhaps the disk check you did wasn't able to quite get everything back to a working state. You may have already done this as part of the upgrade/patch process above.

Finally, ask yourself if you actually know how to accomplish what you are trying to do. Just because you don't know how to do something doesn't mean it is a bug or broken. For example, a woman I used to work with once complained to me that after getting her car windows tinted, her car's back window no longer opened. I asked her how she opened it before. She replied that she forgot. After a quick check of the owner's manual (many technical "secrets" are concealed in things called "manuals"), the window was open.

Don't forget that the search engines are your friend. They don't always give you solutions, but there are a great many solutions out there. Sometimes the item (and not just software, but hardware too) in question has a mailing list or web forums. These often have searchable archives you can try.

Making the bug report

Once you've exhausted the above steps, it's probably time to make a bug report. As frustrating as the bug may be to you, don't try to take your frustrations out in the bug report. Swearing, threatening, and lying never help. The person who receives the bug report may have had nothing to do with causing the bug. The fact you are taking the time to file a good bug report is doing something to make the situation better.

Try to use proper English and proofread what you write. Writing "tHs pro gram cashs where i done clked on there icon with mine moose" will make the bug report reader wonder how seriously to take your report. If you're careless with the language of the report, how accurate is the information in the report?

Stay on topic. The bug report readers really don't want to know what cute thing your dog/cat/child/etc. did today unless it directly relates to the problem at hand (i.e. Your puppy took a whiz on your laptop keyboard and now the laptop doesn't work.). The readers probably have a lot of bad bug reports to wade through. All that fluff takes time away from actually fixing the problems, and isn't getting the problem fixed what's really important?

Details

Probably the most important thing in writing a bug report is detail, detail, detail. Don't generalize. Be as specific as possible. Generalize, and you might get a response like the following taken from the OpenOffice.org mailing list: "Since my crystal ball is on the fritz, it might help if you described the problems you're having." Remember, if the developers, support techs, plumbers, mechanics, doctors and others really had ESP, they would have known what lotto numbers were going to be drawn last night and probably wouldn't be working anymore.

For example, I received, on more than one occasion, a bug report that said the "The Internet doesn't work." This isn't very helpful. If, in one instance, the user had been more specific with "I forgot my e-mail password.", then the problem would have been fixed almost immediately at no cost to the company. As it was, due to some additional bug reporting errors by the user (described below), it took 2 service calls over 3 days for her to get e-mail again.

Don't forget to include all the relevant information, such as operating system, including which version. Don't just say Windows or Linux. Say Windows XP Home SP2 or Ubuntu 7.04. If applicable, what application version are you using. Usually this can be found under Help -> About (sometimes it's "About" then the application name).

If the problem is being reported on-line, think about the main keywords in the title. Don't just put "question" or "problem" as the title, or "Help" or, worse, "Urgent!!!". If you were to search for a solution on the Internet, what keywords would you enter? Be sure and include those keywords in your title and description. Just remember to be specific and not general. This will make it easier for the experts on that facet of the program to find your post and help you. It will also make it easier for someone else with the same problem to find your report and, hopefully, the solution.

Reproducing the Problem

If face to face contact with the person fixing the problem is possible, be prepared to show them the bug. If the tech, developers, plumber, etc. can't see the problem, it is much, much harder to fix. They may notice something you didn't and this gives them the chance to ask questions in order to better define the problem.

In the “The Internet doesn't work.” example above, when the user heard I was in the building, she figured I would need her computer, so she went out to lunch, twice. With such a vague bug report and no chance to ask the user questions, all I could do the first time was check that her web browser worked (usually this is what people mean when they say the Internet isn't working) and list the problem “unable to duplicate” and close it, figuring that since their ISP had problems the day before I got the report, she just never canceled the report even though the problem had already “fixed itself”.

If face to face contact isn't possible, include step by step instructions to replicate the problem. When looking over the instructions, ask yourself, “If I had never seen the computer or the problem, could I duplicate it with these instructions?”. Include any special setup steps or anything else someone would need to do to reproduce the bug.

Don't be afraid to make available some examples. If it's a problem with a document, include a link to a sample document (hopefully with all sensitive information removed) that demonstrates the problem. Some mailing lists automatically strip attachments, so automatically attaching a file to an email is frowned upon.

However, if the problem is intermittent, be prepared to have a lot a patience and time, and sometimes money. Intermittent problems are the hardest problems to solve. Most computer techs I know, including myself, cringe when we hear the word intermittent. Several auto mechanics I've talked to have told me they also cringe when they hear the word intermittent. Don't expect a quick, cheap, or easy fix if the problem is intermittent.

Take a picture

Take a screen shot or picture of the problem or error message. After all, a picture is worth a thousand words. This also prevents mangling of the error message, which can result in the tech looking at the wrong thing if the error message is mangled in just the right way.

In one example, I had to have some scheduled, non-emergency, plumbing work done. When the plumber came, he brought some non-standard correct parts with him. As the parts are a bit unusual, he asked me when his boss, who told him what parts to bring, had been out to look at the piping. I told him his boss had never been there.

He then mumbled in amazement about how good his boss must be to figure what parts were needed from just a phone description. From the plumber's reaction, I think I spoiled a small joke his boss was playing on him when I told the plumber I had e-mailed pictures of the fittings to his boss. As it was, by knowing what parts to bring by having pictures of the problem, I figure I saved over \$80 by the plumber not having to make a second trip. \$80 for taking 3 pictures in the comfort of my home is not a bad rate of pay, plus the problem got fixed sooner. Moral of the story: Don't be afraid to take pictures or screen shots of the problem.

Conclusion

Writing a bad bug report is easy, but it probably won't solve your problem. Even if you do get the problem solved, it will take longer and be more expensive. Writing a good bug report isn't difficult if you keep in mind the pointers above. Yes, it will be a little more work, but that extra work can save you time, money, and frustration. In the end, the extra work of making a good bug report can pay off quite handsomely.

If you would like a longer take on this subject, read Eric S. Raymond & Rick Moen's *How To Ask Questions The Smart Way*: (<http://www.catb.org/~esr/faqs/smart-questions.html>).